Digital Twin and Web Services for Automatic Stacking System in Intelligent Manufacturing

Tianyi Xiang[1], Borui Li[1], Shiqiang Wu[1], Quan Zhang[1,2,*]
[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China
[2]Department of Electrical Engineering and Electronics. University of Liverpool, Liverpool L69 3GJ, UK
*quan.zhang@xjtlu.edu.cn
Tianyi Xiang and Borui Li contribute equally

***Abstract*— The development of manufacturing requires intelligent solutions like digitization, real-time monitoring, and simulation techniques through appropriate digital manufacturing tools. This paper presents the design development of a Digital Twin for a robotic stacking workcell, working principle design, and trajectory optimization. A new programming decision was implemented concerning the different types of workpieces through ABB integrated vision to recognize the types and offset angles of workpieces. This work also involves creating a novel web platform for remote monitoring of the robotic cell. This platform also can send alerts in case of emergency events and provide control access of ABB robotic arm to authorized engineers with the help of RWS, a URL-based command, which cooperate with C# based clients. The goal of this research is to create a workcell-based model for intelligent robotic stacking that incorporates critical concepts such as Cyber-Physical Systems, digitization, data acquisition, continuous monitoring, and intelligent decision in an innovative manner.**

I. INTRODUCTION

In the era of Industry 4.0, manufacturing enterprises are under great pressure to reduce costs and respond rapidly to the changing marketing environment, such as shorter product lifecycle, stochastic orders, individualized product, and the unplanned disruptive manufacturing process [1] [2]. To meet these challenges, the rapid integration of manufacturing resources and real-time decision-making at change scales on a shop floor and even globally has become an urgent issue. For these influences, many efforts have been devoted to developing cyber-physical production systems (CPPS) [3]. The most typical form of CPPS is the digital twin system. [4]

The digital twin is defined as "a formal digital representation of some asset, process or system that captures attributes and behaviors of that entity suitable for communication, storage, interpretation or processing within a certain context" by the industrial internet consortium (IIC) [5] [6]. Some other scholars have given their interpretations in terms of digital twins, yet there is no common definition of this concept, nor the methodology, standards, or norms regarding its implementation [7]. Nevertheless, it is believed that all digital twin systems should have some common characteristics in that it is a two-way mapping of a physical entity (i.e. the physical equipment in the factory shop floor) and a corresponding virtual model (created in computers) [5].

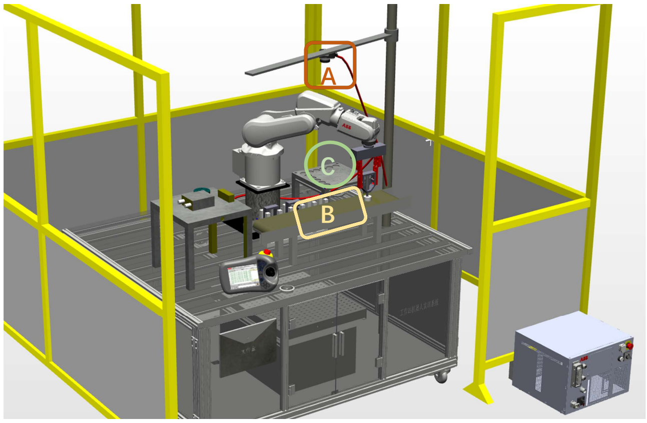


Fig. 1: The Digital Twin of a robotic stacking workcell overview (A: Cognex InSight Camera; B: Recognition-and-Grab Area; C: Checkboard)

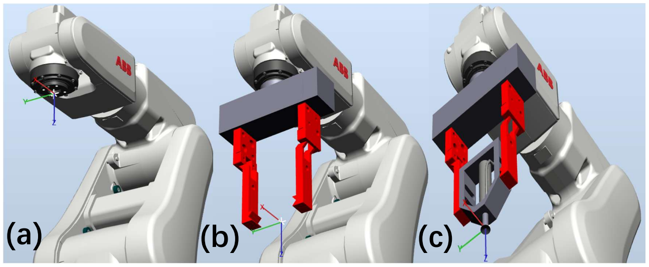
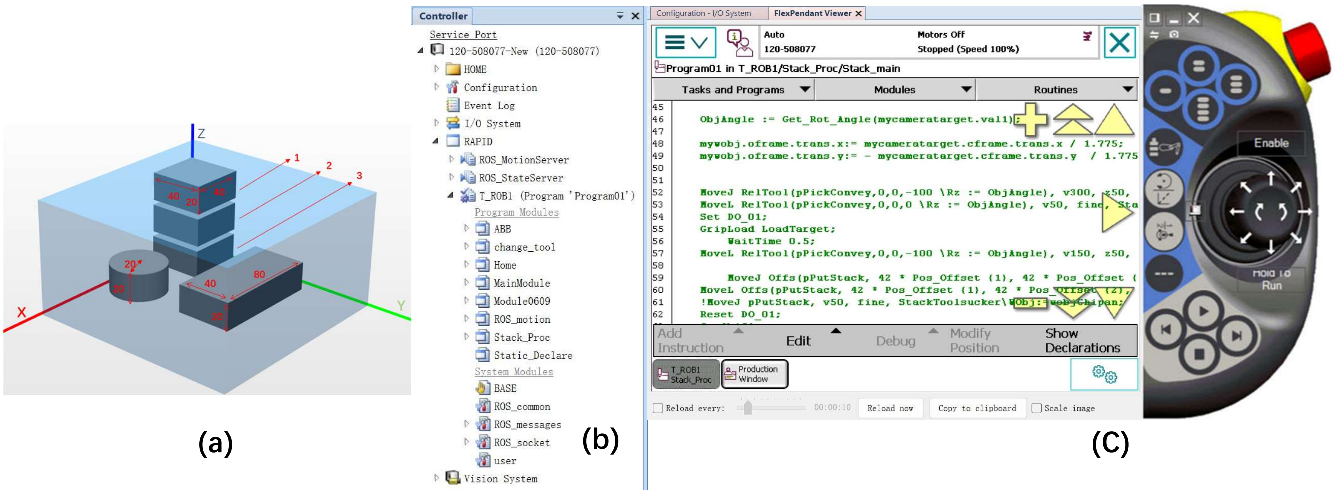


Fig. 2: Tool Center points: (a): Robot tool mounting point; (b): Gripper mounting point; (c): Magnetic sucker

Digital twin technology, as an important means to promote the digital transformation of enterprises and the development of the digital economy, has established a universal theoretical and technical system [8]. However, it is short of practical applications in industrial automation and intelligent manufacturing. Therefore, the practice on implementing the digital twin in real industrial applications forms the subject matter of this paper.

This paper covers the development of a digital twin system for a small-scale robot manufacturing station. As a first attempt, we develop the system based on a simple task in manufacturing applications, i.e. robot grasping. Our



Controller

- Service Port
- 120-508077-New (120-508077)
  - HOME
  - Configuration
  - Event Log
  - I/O System
  - RAPID
    - ROS_MotionServer
    - ROS_StateServer
    - T_ROB1 (Program 'Program01')
      - Program Modules
        - ABB
        - change_tool
        - Home
        - MainModule
        - Module0609
        - ROS_motion
        - Stack_Proc
        - Static_Declare
      - System Modules
        - BASE
        - ROS_common
        - ROS_messages
        - ROS_socket
        - user
  - Vision System

Auto
120-508077

Motors Off
Stopped (Speed 100%)

Program01 in T_ROB1/Stack_Proc/Stack_main

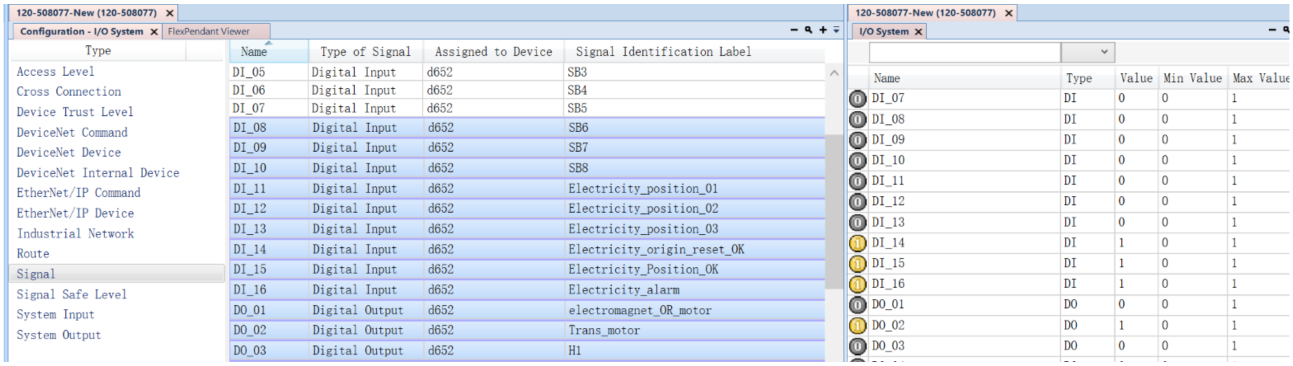
```
    ObjAngle := Get_Rot_Angle(mycameratarget.val1);

    mywobj.oframe.trans.x:= mycameratarget.cframe.trans.x / 1.775;
    mywobj.oframe.trans.y:= - mycameratarget.cframe.trans.y  / 1.775


    MoveJ RelTool(pPickConvey,0,0,-100 \Rz := ObjAngle), v300, z50,
    MoveL RelTool(pPickConvey,0,0,0 \Rz := ObjAngle), v50, fine, Sta
    Set DO_01;
    GripLoad LoadTarget;
        WaitTime 0.5;
    MoveL RelTool(pPickConvey,0,0,-100 \Rz := ObjAngle), v150, z50,

        MoveJ Offs(pPutStack, 42 * Pos_Offset {1}, 42 * Pos_Offset {
    MoveL Offs(pPutStack, 42 * Pos_Offset {1}, 42 * Pos_Offset {2},
    !MoveJ pPutStack, v50, fine, StackToolsucker\WObj:=wobjChipan;
    Reset DO_01;
```

Fig. 3: (a): Target stacking area with multiply coefficients and three kinds of workpieces; (b): Program operation sequence; (c): The program loaded and shown on the robot flex pendant



Type
Access Level
Cross Connection
Device Trust Level
DeviceNet Command
DeviceNet Device
DeviceNet Internal Device
EtherNet/IP Command
EtherNet/IP Device
Industrial Network
Route
Signal
Signal Safe Level
System Input
System Output

Name	Type of Signal	Assigned to Device	Signal Identification Label
DI_05	Digital Input	d652	SB3
DI_06	Digital Input	d652	SB4
DI_07	Digital Input	d652	SB5
DI_08	Digital Input	d652	SB6
DI_09	Digital Input	d652	SB7
DI_10	Digital Input	d652	SB8
DI_11	Digital Input	d652	Electricity_position_01
DI_12	Digital Input	d652	Electricity_position_02
DI_13	Digital Input	d652	Electricity_position_03
DI_14	Digital Input	d652	Electricity_origin_reset_OK
DI_15	Digital Input	d652	Electricity_Position_OK
DI_16	Digital Input	d652	Electricity_alarm
DO_01	Digital Output	d652	electromagnet_OR_motor
DO_02	Digital Output	d652	Trans_motor
DO_03	Digital Output	d652	H1

Name	Type	Value	Min Value	Max Value
DI_07	DI	0	0	1
DI_08	DI	0	0	1
DI_09	DI	0	0	1
DI_10	DI	0	0	1
DI_11	DI	0	0	1
DI_12	DI	0	0	1
DI_13	DI	0	0	1
DI_14	DI	1	0	1
DI_15	DI	1	0	1
DI_16	DI	1	0	1
DO_01	DO	0	0	1
DO_02	DO	1	0	1
DO_03	DO	0	0	1

Fig. 4: Cell signals configuration in the robot controller

proposed digital twin system incorporates tool installation procedures, complete workstation modeling, design working process, and the optimization of stacking trajectory planning. It endeavors to achieve a more efficient visualization-based stacking system by combining C#tutoriala-based client with Unity 3D and Vue-based web design, utilizing Robot Web Service (RWS) – a combination adhering to Rest API and HTTP protocols [9] [10].

II. The Digital Twin of a Stacking Station

The physical characteristics of workcell components are modeled through modern blender tools and visualized through Unity 3D. The DT representation of the virtual modeling workcell is illustrated in Fig.1, performing comprehensive control over the manufacturing process, encompassing the design, planning, testing, collision prevention, fault detection, and monitoring.

In Fig.1.A, the ABB's integrated vision system enables workpiece type recognition and offset angle calculation. The workpieces are transported using a conveyor. The centerpiece of the setup is the 6 DoF articulated arm robot, ABB IRB120 [11], equipped with a gripper that allows seamless tool changes, including a magnetic sucker tool used to pick up and place workpieces on the checkboard. Moreover, the CAD design includes supplementary tools such as grinding and polishing tools."

In Fig.2, three different tool center points (TCP) were displayed for: (a): the robot's tool mounting point, (b): the gripper mounting point, and (c): the magnetic sucker tool, which is used to grab the iron workpiece.

III. Working Process of the Workcell

A. Robot programming design with IO signals configuration

Before creating the Digital Twin, ABB's product, RobotStudio was used to present the robotic workcell, generate collision-free tool paths, and program the robot [11]. The software also considers sensor data, analysis of motion results, and evaluation of robot energy consumption.

Events have been established using I/O signals that serve as communication channels between the controller and external equipment. For instance in Fig.4, DO_02 signal enables activating the conveyor.

The programming included robot movements for tool changes and picking up and placing workpieces between the gravitational aligning table, fixture system, collision checks, and outfeed conveyor.

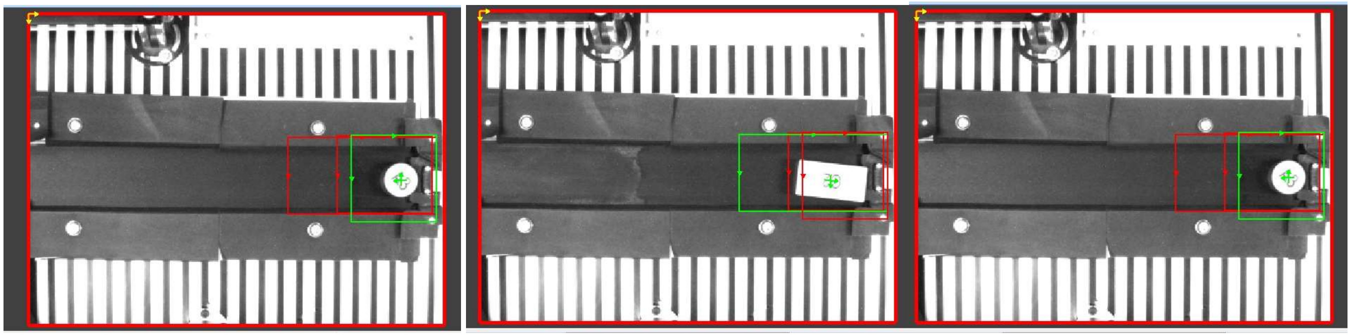

(a) (b) (c)

Fig. 5: Camera Detection and offset angle recognition (a): circle workpiece; (b): rectangular workpiece; (c): circular workpiece

The robotic stacking sequence is illustrated in Fig6 to certify all workpieces are entirely packed. There are three working stages:

- *In the 1st stage*, the workpieces are sent through the conveyor and recognized by the Cognex Insight Camera, then stop immediately when the infrared distance sensor at the end of the conveyor is triggered to ensure a relatively precise position and orientation before it is packed. However, small position errors could be compensated depending on the offset angle given by integrated vision, which will be discussed later.
- *Moving on to the 2nd stage*, inspired by the concept introduced by [12], the workpiece is brought to the designated checkboard depicted in Fig.3. Here, the optimal orientation for stacking is calculated. Initially, a 3 × 3 × 3 occupation-record field is established to represent the stacking area. Subsequently, the robot assigns a score to each potential stacking point based on the types of workpieces. Subsequently, the robot multiplies three corresponding coefficients along the X, Y, and Z axes according to the desired stacking priority to identify the best area for stacking. For example, Fig.3.(a) illustrates the scoring mechanism along the Z-axis, generating three different scores while evaluating the ideal position for stacking the square workpiece.
- *In the 3rd stage*, the robot waits for a new workpiece transported by the conveyor or retrieves the magnetic sucker back and goes safety point.

Similar to [13], sensor signals were configured in the base of workstation logic and synchronized to the virtual robot controller IO environment shown in Fig.4.

After designing all robot actions the virtual representation could show programs, each target point, orientation, programming sequence, and communication within the upper computer. The robot stacking sequence, communication programs, and robot movements are illustrated in Fig.3.(b).(c).

Finally, the well-integrated data could be sent through RWS (Robot Web Service), including Robot joint data, DO signals, and production information to the back-end client, which will be explained later.

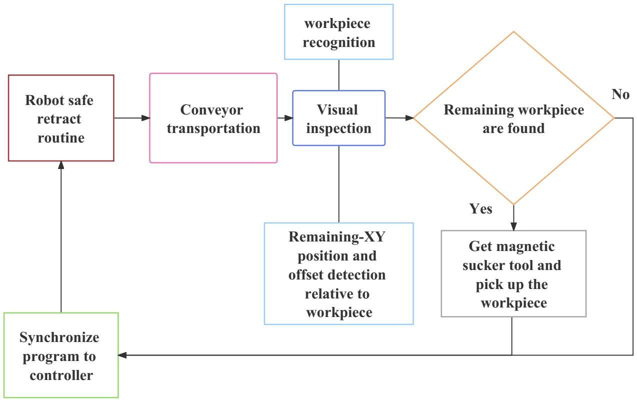

workpiece recognition
Robot safe retract routine
Conveyor transportation
Visual inspection
Remaining workpiece are found
No
Yes
Remaining-XY position and offset detection relative to workpiece
Get magnetic sucker tool and pick up the workpiece
Synchronize program to controller

Fig. 6: Comprehensive representation of the stacking workflow

B. Vision Recognition and Offset Angles

Due to the uncertainty in the types and angles of workpieces transported by the conveyor, this study focuses on workpiece type and offset angle recognition. In Fig.5.(a).(b).(c), three types of workpieces are initially identified. Then, an automated process calculates the significant offset angle, which is represented as a green crosshair cursor. For instance, in Fig.5.(b), when the rectangular workpiece is not aligned straight with the conveyor, the offset angle can be calculated and subsequently incorporated into the robot's "MOVEL" command using the "Offs" function while attempting to grasp the workpiece. As a result, the robot can effectively optimize its routines and achieve central symmetry positioning for the workpiece.

IV. THE OPTIMIZATION OF ROBOT STACKING TRAJECTORY PLANNING

To optimize the energy consumption of the entire processing procedure, four steps are designed as follows:

- MoveJ to the prepared grab position.
- MoveL to grab the workpiece.
- MoveL to the safety point above the stacking area.
- MoveL to place the workpiece.

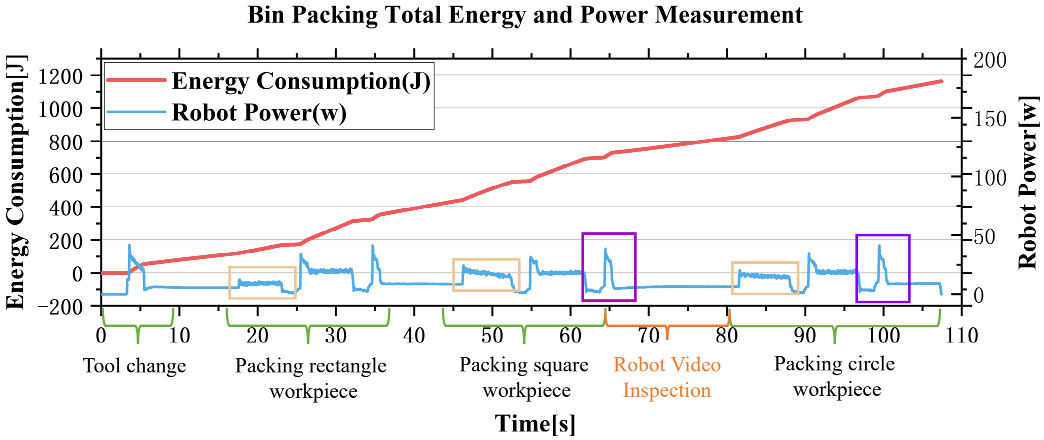


Fig. 7: Records of the total energy (marked in red) and total power (marked in blue) of stacking sequence processing three kinds of workpieces (rectangle, square, circle) with tool change and robot video inspection

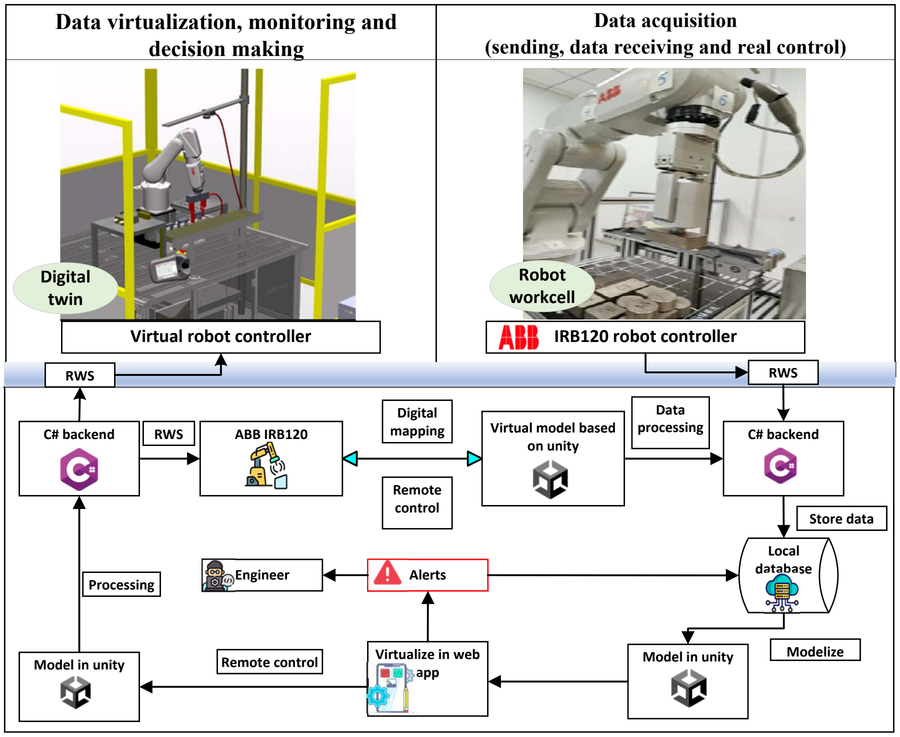
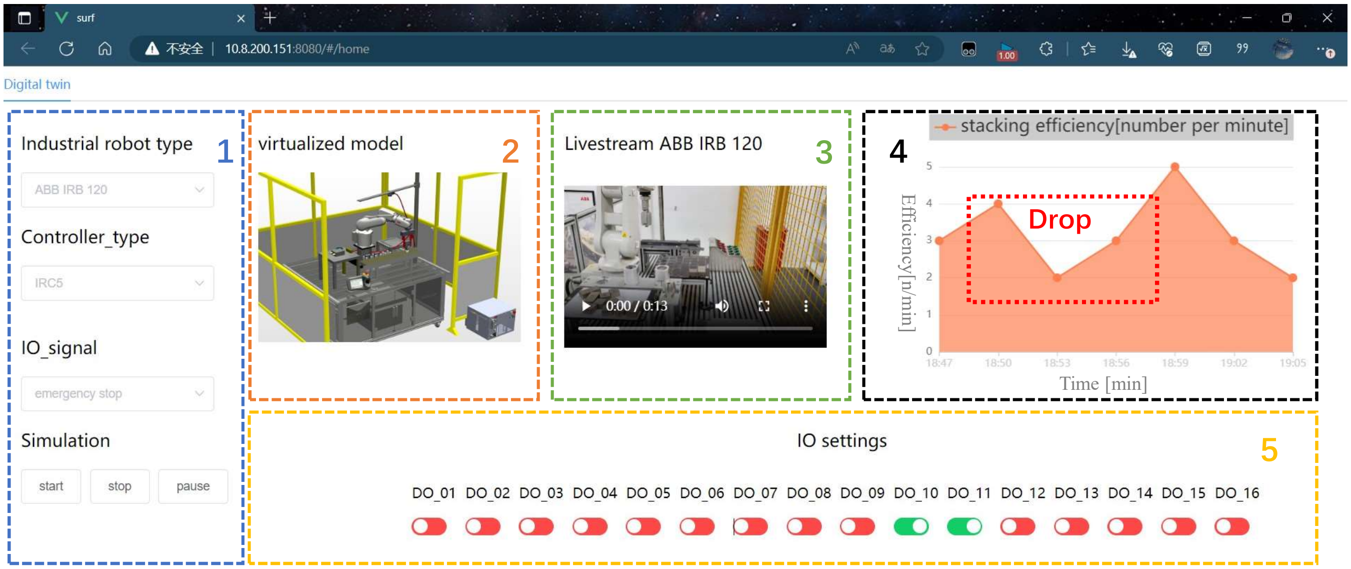


Fig. 8: Web-based robotic stacking system platform for monitoring and remote control

To ensure minimum energy and time consumption, MoveJ manipulates all joints together, enabling the robot to move in the fastest and unique routine, albeit with partial lack of controllability [14]. However, the reliability and stability of transporting the workpiece are also crucial. Therefore, the workcell is designed to use ”MoveL” for linear workpiece transportation, following a straight line path between the current position and the destination [14].

The overall performance of the workcell is monitored and recorded in Fig.7. This monitoring includes capturing the real-time Power and Total Energy curve during typical operating conditions. These conditions involve stacking each type of workpiece (rectangle; square; circle), tool change, and robot video inspection. The optimized results of the designed optimized stacking trajectory planning can be observed as follows:

- The smoothness of the energy curve shown in the red curve reflects the robot’s ability to maintain a steady energy consumption during the stacking process. This indicates that it avoids unnecessary energy waste and follows a safe routine for transportation.
- Secondly, the power curve consistently stays within



Industrial robot type: ABB IRB 120

Controller_type: IRC5

IO_signal: emergency stop

Fig. 9: Initialize panel (marked in blue) and 3D model of the workplace (marked in red) and live-stream of 3D model of the workplace and working efficiency of the stacking (marked in black) and the control panel of IO settings (marked in yellow)

the designated orange box region, maintaining a low level with minimal power peaks. This allows the robot to avoid abrupt acceleration or deceleration during the stacking process. As a result, the robot can efficiently complete the stacking task while simultaneously reducing peak energy consumption and minimizing the overall system load.

V. WEB BASED PLATFORM FOR ROBOTIC STACKING REMOTE CONTROL AND MONITORING

This project aims to implement a web-based monitoring and remote control system to keep pace with Industry 4.0. The system utilizes Web services with the RWS (Robot Web Services) [9]protocol to acquire real-time information from the actual workplace. RWS is an HTTP-based protocol capable of retrieving current data through URLs and receiving data in XML or JSON formats. Additionally, the platform allows users to remotely control robotic arms by sending desired locations for precise positioning. The received data is processed and stored in two databases: the first database stores production logs together with error logs. In contrast, the second database stores the current IO signals value, displaying the machine status.

The working process of the web-based platform is illustrated in Figure 8. For data acquisition, as shown on the right, the C# [15]-based client automatically sends requests based on RWS [9] every 1 second in JSON format. After receiving the data, the client processes and stores it in the local database, and it collaborates with Unity, where the model moves according to the received six-axis data. Moreover, in case of critical events, such as axis angles exceeding the safe range or the gripper getting stuck, the platform alerts engineers online to ensure the robot's safety.

For remote control, engineers can manipulate the model by dragging buttons to adjust the six axes. The data is then sent to the client, which forwards the URL to update the angles. This results in synchronized movement of both the virtual and actual models. As exhibited in Fig.9, the following will represent a few functions that the web-based platform can represent.

- Func.1: The 1th section blue section.Robot and Controller Selection. Machine Control. The platform allows users to control the entire machine's power, offering remote start, pulse, and stop functions. This feature ensures precise control over production processes for improved efficiency.
- Func.2, the 2nd orange section, modeled Workplace Visualization. The platform leverages Unity to provide users with a 3D modeled representation of their workplace. This visualization aids in better understanding the layout and arrangement of robotic systems within the factory floor.
- Func.3, the 3rd green section.Robot and Controller Selection. Real-Time Workplace Status. Users can monitor the real-time status of the workplace through the platform. This includes information on robot movements, production progress, and error notifications, enabling swift intervention when required.
- Func.4: The 4th section.Robot and Controller Selection. Efficiency Analysis and Charts. Data collected via RWS [9]is used to generate efficiency charts, providing valuable insights into production performance. Users can analyze these charts to identify bottlenecks, optimize processes, and enhance overall productivity.
- Func.5: The 5th yellow section.Robot and Controller Selection. DO System Management. Users can control the entire DO system, regulating the activation and deactivation of various components such as conveyors and

electromagnets. This allows for dynamic adjustments to the production line configuration.

VI. Remote Monitoring and Control

This project presents a web-based remote control system for an ABB robotic arm, illustrated in Fig.8, integrating the RWS (Robotic Web Services) [9]protocol and RAPID language. The system offers two types of control: local control and remote control. When operating in remote control mode, the robotic arm's RAPID program detects user commands from a C#tutoriala-based client and creates a variable called ”location-R.” The client will send the actual location of the targeted position by using a specific format. This variable enables the robotic arm to be precisely controlled by the user, either by dragging sliders or using the control panel to manipulate the six-axis degree. For the dragging method, one drag will cause the J1 to J6 to move a specific degree, and the data will be received in the client and processed. After that, it will be sent to the actual ABB robotic arm as formulated data. In addition, this platform can also open/close the electromagnet, open/close the conveyor, open/close the alarm, and open/close the gripper. These can be realized by changing the value of IO systems' values. RWS [9] provides the URL that can monitor and change the value of the DO signals, making remote control a realized dream.

VII. Conclusion

This paper has successfully developed a Digital Twin for a robotic stacking workcell. Our contribution includes the following aspects:

- Through blender tools and visualizes it through Unity 3D, construct the digital twin robotic stacking workcell and remote monitoring and control platform communicating by RWS between Vue based web platform and the c# based client.
- Employed robot programming through tool assembly and IO signals configuration based on TCP(Tool center points) and working principles.
- Implemented a new working process concerning the different types of workpieces through ABB integrated vision to recognize the types and offset angles while minimizing energy consumption and ensuring reliability through monitoring energy and power curve.

References

[1] H. Zhang, Q. Yan, and Z. Wen, “Information modeling for cyber-physical production system based on digital twin and AutomationML,” *The International Journal of Advanced Manufacturing Technology*, vol. 107, no. 3, pp. 1927–1945, 2020.

[2] A. Martins, H. Costelha, and C. Neves, “Supporting the Design, Commissioning and Supervision of Smart Factory Components through their Digital Twin,” in *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2020, pp. 114–119.

[3] Y. Zhang, C. Qian, J. Lv, and Y. Liu, “Agent and Cyber-Physical System Based Self-Organizing and Self-Adaptive Intelligent Shopfloor,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 737–747, 2017.

[4] R. V. Yohanandhan, R. M. Elavarasan, P. Manoharan, and L. Mihet-Popa, “Cyber-physical power system (cpps): A review on modeling, simulation, and analysis with cyber security applications,” *IEEE Access*, vol. 8, pp. 151 019–151 064, 2020.

[5] J.-G. Duan, T.-Y. Ma, Q.-L. Zhang, Z. Liu, and J.-Y. Qin, “Design and application of digital twin system for the blade-rotor test rig,” *Journal of Intelligent Manufacturing*, vol. 34, no. 2, pp. 753–769, 2023.

[6] D. Ramos, G. Gonçalves, R. Faria, and M. P. Sanches, “Building a Drone Operator Digital Twin using a Brain-Computer Interface for Emotion Recognition,” in *2021 20th International Conference on Advanced Robotics (ICAR)*, 2021, pp. 824–829.

[7] F. Tao and Q. Qi, “Make more digital twins,” *Nature*, vol. 573, no. 7775, pp. 490–491, 2019.

[8] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, “Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101837, 2020.

[9] “Robot Web Services: Introduction,” https://developercenter.robotstudio.com/api/rwsApi/index.html.

[10] “Vue.Js,” *Wikipedia*, 2023.

[11] “ABB,” *Wikipedia*, 2023.

[12] Y. Li, K. Wang, Y. Zhai, and Q. Zhang, “Automatic Stacking System based on ABB Robots and Digital Twin Monitoring,” 2021.

[13] L. Stan, A. F. Nicolescu, C. Pupăză, and G. Jiga, “Digital Twin and web services for robotic deburring in intelligent manufacturing,” *Journal of Intelligent Manufacturing*, vol. 34, no. 6, pp. 2765–2781, 2023.

[14] Y. Bai, X. Yang, and W. Huang, “Research on trajectory optimization of handling robot based on abb,” in *2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 2020, pp. 109–113.

[15] “C# Tutorial (C Sharp).”